

المادة: مخبر نظم التشغيل BOSL501

الوظيفة S24

الموعد النهائي لتسليم الوظائف 2024/12/ 25

موضوع الوظيفة	عملي على إدارة الإجراءات ومزامنتها
رقم الوظيفة	1
منسق المقرر	م.ماجدة أسعد
أستاذ المقرر	م.ماجدة أسعد
العلامة العظمى	100
تاريخ التسليم	25/12/2024
عدد الطلاب	5 طلاب كحد اقصى
الموارد المطلوبة	نظام مفتوح المصدر -/ مترجم gcc

الهدف من الوظيفة والفائدة المرجوة للطلاب من إنجازها

تهدف هذه الوظيفة إلى تعلم كيفية إنشاء الاجرائيات والنياسب في Linux والعمل معها ومعالجتها واستخدام الاشارات بينهما. و حل المسائل المستعصية في عمل ومزامنة الإجراءات والنياسب عند استخدام موارد مشتركة باستخدام السيمافورات.

الأوامر والاستدعاءات المستخدمة: مكتبات POSIX حصراً وليس Microsoft.

ملاحظات

1. عدل ملف الوظيفة لكتابة تقرير توضح به البرامج وكل امر يتم استخدامه والغاية منه، وخطوات الحل، مع صور لنتائج الخرج لكل طلب.
2. ارفاق الملفات المصدريّة source code والتنفيذية مع التقرير.
3. حمل على موقع AMS التقرير بصيغة ملف pdf يحمل الاسم:

السؤال الأول Thread-Signals-Process (35 علامة)

اكتب برنامج C يقوم بما يلي:

- a. تقوم الإجراءية الام (البرنامج الرئيسي) بإنشاء إجراءية ابن فور تشغيلها. (3 علامة)
- b. تنتظر الإجراءية الأم لحين انتهاء الابن. (3 علامة)
- c. تنشئ الإجراءية الابن 5 نياسب . (5 علامات)
- d. تقوم النياسب بارسال إشارة عشوائية signal للإجراءية الام (SIGCHLD, SIGINT, SIGTRAP,) (10 علامات) (SIGUSR1, SIGKILL).
- e. عند ارسال إشارة رقم 9 من احدى نياسب الإجراءية الابن للإجراءية الام, تعيد الإجراءية الام إشارة رقم 17 لخروج الجرائية الابن. (6 علامات)
- f. تنتظر الإجراءية الابن, التي أنشئت النياسب, انتهاء جميع النياسب. (5 علامات)
- g. اظهر تأثير مشاركة وحدة المعالجة المركزية الخاصة بهم ومعلومات الذاكرة الرئيسية والافتراضية, وحالاتهم في حال تم تنفيذ البرنامج على معالج بنواة واحدة (core) او عدة نوى داخلية للمعالج, مع شرح الخرج الناتج. (3 علامات)

السؤال الثاني: Threads-PIPES (25 علامة)

اكتب برنامج بلغة C ينشئ جلسة محادثة ثنائية الاتجاه بين نيسيين.

- a. أنشئ البرنامج الرئيسي.
- b. أنشئ النياسب اللازمة, وعددها اثنان. بحيث كل نيسب يكون مرسل ومستقبل. (5 علامات)
- c. انشاء الانابيب اللازمة PIPE لانجاز عملية القراءة والكتابة وباتجاهين. (20 علامة)

السؤال الثالث Process Synchronizing (40 علامة)

مسألة عشاء الفلاسفة هي إحدى المسائل الشهيرة في تزامن العمليات وتجنب حالات الاقفال المتبادل (Deadlock) وحالات الجوع (Starvation). طُرحت هذه المسألة كتمثيل لمشكلة تزامن العمليات، حيث يكون هناك عدد من العمليات (أو الفلاسفة) تحتاج للوصول إلى موارد مشتركة (شوكة لكل يد) بطريقة لا تؤدي إلى تعارض أو توقف العملية كليًا.

وصف المسألة

لنفترض أن هناك خمسة فلاسفة يجلسون حول طاولة دائرية لتناول الطعام. أمام كل فيلسوف يوجد صحن من الطعام، وبين كل فيلسوفين توجد شوكة واحدة. ولكي يستطيع الفيلسوف أن يأكل، يحتاج إلى شوكتين (واحدة من اليسار وواحدة من اليمين). بمجرد الانتهاء من تناول الطعام، يعيد الفيلسوف الشوكتين إلى الطاولة ويعود إلى التفكير. تهدف المسألة إلى تحقيق ما يلي:

١. لا يستطيع الفيلسوف الأكل إلا إذا كان لديه شوكتان.
٢. لا يمكن لأكثر من فيلسوفين مجاورين أن يأكلوا في نفس الوقت.
٣. يجب ضمان عدم حدوث تعارض يؤدي إلى توقف جميع الفلاسفة عن الأكل.
٤. تجنب حدوث حالات الجوع، بحيث يحصل كل فيلسوف على فرصته في الأكل.

المطلوب:

تعديل برنامج عشاء الفلاسفة بحيث يكون هناك 5 فلاسفة ولكن 3 شوكات فقط، سنحتاج إلى بعض التعديلات على طريقة تزامن الفلاسفة. في هذا السيناريو، نريد أن نضمن أن الفلاسفة يمكنهم الوصول إلى الشوكات المتاحة، ولكن نظرًا لأن عدد الشوكات أقل من عدد الفلاسفة، قد يضطر الفلاسفة إلى الانتظار حتى تتوفر شوكتان للبدء في الأكل.

الخطوات:

١. أنشئ 3 سيمافورات للشوكات، والتي تحدد ما إذا كانت الشوكة متاحة أم لا. (10 علامات)
٢. أنشئ 5 إجراءات أبناء تمثل الفلاسفة. (5 علامات)
٣. يحتاج الفيلسوف إلى التأكد من وجود شوكتين متاحيتين قبل أن يحاول الأكل. إذا لم يجد شوكتين، فسيتعين عليه الانتظار حتى يتم تحرير شوكات كافية. (5 علامات)
٤. أنشئ التوابع اللازمة لتمثيل حالة التفكير وحالة الأكل. (10 علامات)
٥. استخدم استدعاءات النظام اللازمة لحل المسألة، وتحقيق التزامن ومنع الاقفال المتبادل بينهما. (10 علامات)

انتهت الأسئلة